Copy No._____

Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Final Report for Data Processing and Analysis Support to Measure Active Signal Levels

A Call-up under the IMPACT Standing Offer

Joe Hood, Brad Glessing, Olivier Beslin and Derek Jury
MacDonald Dettwiler and Associates

MacDonald Dettwiler and Associates
Suite 60, 1000 Windmill Road
Dartmouth, NS
B3B 1L7

Contract Number: W7707-03-2304

Contract Scientific Authority: James. A. Theriault (902) 426-3100 ext 376

The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

Terms of Release: *The information contained herein is proprietary to Her Majesty and is provided to the recipient on the understanding that it will be used for information and evaluation purposes only. Any commercial use including use for manufacture is prohibited. Release to third parties of this publication or information contained herein is prohibited without the prior written consent of Defence R&D Canada.*

Defence R&D Canada – Atlantic

Contract Report
DRDC Atlantic CR 2004-075
April 2004

Canada

Copy No: _____

Final Report for Data Processing and Analysis Support to Measure Active Signal Levels

A Call-up under the IMPACT Standing Offer

Joe Hood
MacDonald Dettwiler and Associates Ltd.

Brad Glessing
MacDonald Dettwiler and Associates Ltd.

Olivier Beslin
MacDonald Dettwiler and Associates Ltd.

Derek Jury
MacDonald Dettwiler and Associates Ltd.

Contract # W7707-03-2304

Contract SA: James A. Theriault

The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

Terms of release: The information contained herein is proprietary to Her Majesty and is provided to the recipient on the understanding that it will be used for information and evaluation purposes only. Any commercial use including use for manufacture is prohibited. Release to third parties of this publication or information contained herein is prohibited without the prior written consent of Defence R&D Canada.

Defence R&D Canada – Atlantic

Contract Report

DRDC Atlantic CR 2004-075

April 2004

105-04555/

Author

Joe Hood

Approved by

Jim Theriault

Scientific Authority

Approved for release by

Kirk Foster

DRP Chair

The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada

Terms of release: The information contained herein is proprietary to Her Majesty and is provided to the recipient on the understanding that it will be used for information and evaluation purposes only. Any commercial use including use for manufacture is prohibited. Release to third parties of this publication or information contained herein is prohibited without the prior written consent of Defence R&D Canada.

© Her Majesty the Queen as represented by the Minister of National Defence, 2004

© Sa majesté la reine, représentée par le ministre de la Défense nationale, 2004

Abstract

This report documents the work done to convert recorded multistatic sonar trial data to received signal level measurements. The signals were received at three hydrophones placed on the target submarine during Q246. The work was broken down into a number of tasks with several tasks involving the generation of reusable software components and analysis processes, allowing future work to be performed more efficiently. The received signals were digitized using the Environmental Acoustic Data Acquisition System (EADAQ) and processed using existing components from the Signal Processing Package (SPPACS) software suite. The Software Tools for Analysis and Research (STAR) Suite was then used to perform data analysis including test, calibration and measurement tasks. Analysis logs were imported into Microsoft Excel for archiving and final analysis of the results. Considerable efficiency was realized by using common databases, software and processes that Defence Research and Development Canada (DRDC) plans to reuse to perform similar tasks in the future. This document describes the processing and software environment used for the project.

Résumé

Le présent rapport décrit les travaux effectués en vue de convertir des données d'essais multistatiques brutes en mesures de niveaux de signaux reçus. Dans ce cas, les signaux ont été reçus à trois hydrophones placés sur le sous-marin cible durant Q246. Les travaux ont été subdivisés en un certain nombre de tâches, plusieurs tâches nécessitant la production d'éléments logiciels et de procédés d'analyse réutilisables qui visent à améliorer l'efficacité des travaux futurs. Les signaux reçus ont été numérisés au moyen du système d'acquisition de données acoustiques environnementales (EADAQ) et traités au moyen d'éléments faisant partie de la suite logicielle de l'ensemble de traitement de signaux (SPPACS). La suite des outils logiciels d'analyse et de recherche (STAR) a ensuite servi à effectuer une analyse de données comprenant des tâches d'essai, d'étalonnage et de mesure. Les journaux d'analyse ont été importés dans Microsoft Excel aux fins d'archivage et d'analyse finale des résultats. Il a été possible d'obtenir un rendement remarquable en recourant à des bases de données, à des logiciels et à des procédés communs que R & D pour la défense Canada (RDDC) projette de réutiliser dans l'avenir pour effectuer des tâches semblables.

This page intentionally left blank.

Executive summary

Background

DRDC Atlantic has ongoing research efforts in active sonar. Specifically, projects such as the IMPACT (Integrated Multistatic Passive Active Concept Testbed) and TIAPS (Towed Integrated Active-Passive Sonar System) generate a significant volume of acoustic data. In analyzing and developing and understanding of the performance of demonstration systems, it is critical that targets be instrumented with both acoustic and non-acoustic sensors.

By analyzing the time series of the acoustic sensors and combining with positional information, the impact of the environment on the active sonar detection capability can be assessed.

Results

This effort concentrated on developing a processing capability and generating received levels from a 1998 sea trial involving both a Maritime Patrol Aircraft (MPA) using the IMPACT processor and CFAV Quest using a towed active sonar system. This report focuses on the software development. The received level analysis for the trial has been delivered to the contract authority under separate cover.

Significance

The enhancements to the analysis packages STAR (Software Tools for Analysis and Research) and SPPACS (Signal Processing Package) greatly improve DRDC Atlantic's ability to analyze acoustic data, with supporting non-acoustic data, in a timely manner.

Future Plans

The suite of tools developed under this contract will be used for further studies of active sonar performance in order to validate both IMPACT and TIAPS.

Hood, Joe, Brad Glessing, Olivier Beslin, and Derek Jury, 2004. Final Report for Data Processing and Analysis Support to Measure Active Signal Levels, A Call-up under the IMPACT Standing Offer,. DRDC Atlantic CR 2004-075. Defence R&D Canada – Atlantic

Sommaire

Contexte

RDDC Atlantique mène actuellement des recherches sur les sonars actifs. Plus précisément, des projets comme IMPACT (banc d'essai multistatique intégré actif/passif) et TIAPS (sonar actif-passif intégré remorqué) génèrent un volume important de données acoustiques. Pour qu'il soit possible d'analyser, d'améliorer et de comprendre le rendement des systèmes de démonstration, il est primordial que les cibles soient dotées à la fois de capteurs acoustiques et non acoustiques.

L'analyse de la série temporelle des capteurs acoustiques et la combinaison avec l'information de position permettent d'évaluer l'impact de l'environnement sur la capacité de détection du sonar actif.

Résultats

Les travaux ont essentiellement porté sur le développement d'une capacité de traitement et sur la production des niveaux reçus au cours d'un essai en mer de 1998 qui faisait appel à un aéronef de patrouille maritime (APATMAR) utilisant le processeur IMPACT et au NAFC Quest utilisant un sonar actif remorqué. Le rapport met l'accent sur la production de logiciel. L'analyse des niveaux reçus durant l'essai a été soumise à l'autorité contractante sous pli séparé.

Portée

L'amélioration des systèmes d'analyse STAR (outils logiciels d'analyse et de recherche) et SPPACS (ensemble de traitement de signaux) améliore considérablement l'aptitude de RDDC Atlantique à analyser les données acoustiques, avec des données non acoustiques complémentaires et dans les délais requis.

Recherches futures

La suite des outils produits dans le cadre de ce contrat servira à d'autres études de rendement du sonar actif destinées à la validation d'IMPACT et de TIAPS.

Hood, Joe; Glessing, Brad; Beslin, Olivier; et Jury, Derek; 2004, Rapport final sur le traitement des données et le soutien à l'analyse pour la mesure des niveaux de signaux actifs, Commande subséquente à l'offre permanente IMPACT, RDDC Atlantique CR 2004-075, R & D pour la défense Canada – Atlantique

Table of contents

Abstract i

Executive summary iii

Sommaire iv

Table of contents v

List of figures viii

1. Introduction 1
   - 1.1 Overview 1
   - 1.2 Background 1
     - 1.2.1 STAR 1
     - 1.2.2 SPPACS 2

2. Work Overview 4

3. Digitize Data 6

4. Data Processing and Analysis 7
   - 4.1 NAD Formatting 7
   - 4.2 Data Processing 7
     - 4.2.1 Signal Level Processing 8
   - 4.3 Calibration and Testing Methodology 8
     - 4.3.1 Time domain energy 9
     - 4.3.2 Discrete Fourier Transform Conventions 9
       - 4.3.2.1 First Convention 9
       - 4.3.2.2 Second Convention 10
     - 4.3.3 Frequency Domain Energy 10
       - 4.3.3.1 SPPACS Expression 10
       - 4.3.3.2 IDL Expression 11
     - 4.3.4 Calibration 11
   - 4.4 Data Analysis 11

5. Developed Software Components 13
5.1 Suspend Analysis 13
5.1.1 Requirements information 14
5.1.2 Software Design information 14
5.1.3 Test Description 15
5.1.4 Test Report 16
5.1.5 User Instructions 16
5.2 Automated Reanalysis 16
5.2.1 Requirements 17
5.2.2 Software Design 17
5.2.2.1 Modules 17
5.2.3 Test Description 19
5.2.4 Test Report 19
5.2.5 User Instructions 19
5.3 Hardcopy Generation 19
5.3.1 Requirements 20
5.3.2 Software Design 20
5.3.3 Test Description 21
5.3.4 Test Report 21

6. Software Maintenance and Design Improvement 22
6.1 Window Resizing 22
6.1.1 Software Design information 22
6.1.2 Test Description 22
6.2 Capture Window Close 23
6.2.1 Software Design information 23
6.2.2 Test Description 23
6.3 IDL API Document Generation 23
6.3.1 Requirements Information 23
6.3.2 Software Design information 24
6.3.3 Test Report 26
6.3.4 User Instructions 27

7. Data Analysis Recommendations 28

8. Maintain Configuration Management 29

9. Track Software Issues 30

Bibliography 31

List of symbols/abbreviations/acronyms/initialisms 32

Distribution list 34

List of figures

Figure 1. Suspend Analysis – Save Analysis Data 14

Figure 2. Suspend Analysis - Access Analysis Data 15

Figure 3. Reanalysis Activity Diagram 18

Figure 4. Hardcopy Generation Sequence Diagram 20

Figure 5. Sample IDL with Tags 25

Figure 6. idl_src_parse HTML Output 26

List of tables

Table 1. Contract Requirements 5

Table 2. Test Results – Suspend Analysis 16

Table 3. Test Results – ETI and Power Spectra 19

Table 4. Test Results – Hardcopy Generation (ETI, Power Spectra, Tactical Plot, BBTR) 21

Table 5. Distribution Issue Summary (18/03/2004) 30

1. Introduction

This final report outlines the work done under the Research, Development and Engineering Services on Behalf of the Integrated Multistatic Passive Active Concept Testbed (IMPACT) Regional Individual Standing Offer (RISO) No. W7707-021973/001/HAL, Data Processing and Analysis Support to Measure Active Signal Levels Call-Up Requisition No. W7707-03-2304. This work was performed at DRDC Atlantic under the direction of Scientific Authority (SA), Jim Theriault from approximately December 2003 to March 2004.

1.1 Overview

The objective of this call-up was to provide data processing and analyses support as required to measure active pulse signal levels at the target of interest and associated parameters for specific multistatic trial data from Q246. The trial data, consisting of multistatic active signal recordings measured at the target of interest, was recorded on three hydrophones over approximately 5 days. The data was recorded on Metrum tapes, which could be replayed as analogue data.

Q246 was a joint Canadian - United Kingdom multistatic Low Frequency Active (LFA) trial that was conducted in the military operation areas south of Halifax, Nova Scotia. A number of surface ships, submarines and aircraft participated in the trial. Both towed arrays and passive sonobuoys were used as multistatic receivers and a number of towed LFA sources were employed. One submarine was fitted with three passive hydrophones that recorded the LFA pulses incident on the target. This project is intended to measure the absolute energy level of the incident LFA pulses.

1.2 Background

1.2.1 STAR

Most of the software, developed under this contract, involved enhancements to the STAR suite. The primary objectives of the STAR Suite [1] are:

- Provide scientific grade analysis tools that allow for efficient, detailed quantitative and qualitative analysis of a data set.
- Support synergy between DRDC groups and the Department of National Defence (DND) by providing a common software base for analysis. This synergy encourages inter-group communication and simplifies user training, analysis process development, documentation and data portability.

- Support cost and analysis efficiency by providing software reuse and common tools and data formats. Examples of efficiency would be using the output of analysis from one group to feed the inputs of another, or using common software components to lower development cost of several custom analysis tools.

All STAR components are currently implemented using the Interactive Data Language (IDL), though the design is not restricted to IDL. The name STAR reflects the generic nature of the software. Applications in the STAR suite are built using a combination of reusable and custom components that meet the requirements of each task. The layered design and common components allow for rapid and logical development of new capabilities. Though currently focused on sonar data processing and analysis, the tools are capable of expanding to meet other analysis and research requirements.

The STAR suite is configuration controlled using the concurrent versioning system (CVS) and issue and enhancement idea tracking is affected using the Bugzilla issue tracking software. CVS is a repository that allows developers to check-in revisions to software and documentation where they are archived in a common database. The tool allows all previous versions of the software to be maintained and aids resolution of new issues, while ensuring that current builds of the software are readily accessible to users and developers alike. Bugzilla is a web accessible database that offers both user and developer input to issues, priorities and solutions. It provides coherent tracking and recording of an issue over its entire lifecycle.

STAR components are documented in a combination of formats, each with their own purpose. Microsoft Word documents are maintained that describe functionality and algorithms of components. These are primarily intended for the end user. Enterprise Architect (EA) files are maintained that document software design, interaction and dependencies. EA design information is intended primarily for developers. HTML library documentation is being developed that provides automatic extraction of the routine's Application Program Interface (API), purpose and description. This documentation is maintained to assist developers familiarize themselves with the existing libraries and components and is intended to support and encourage software reuse. Some users may also wish to refer to this information for use in their own custom applications.

1.2.2 SPPACS

The Signal Processing Package (SPPACS) [2] was used to perform the required signal processing for this contract. SPPACS is a group of software programs that is based on the C programming language and is implemented on Linux based Personal Computers (PC). Each program provides a specific processing function and a series of programs can be chained together to create a custom-processing stream using the command line or scripts. The output from SPPACS is stored in DREA formatted data files. SPPACS has slowly evolved to its present day state due to the efforts of several MacDonald Dettwiler and Associates Ltd. (MacDonald Dettwiler) personnel over the last 4 years.

SPPACS has been used to perform a number of mid-trial and post-trial processing functions such as the post-trial study of multistatic trial data and the mid-trial analysis of the Q246 sonobuoy test trial. SPPACS only performs data manipulation and does not provide an interface to examine the results. The processed data output is often imported into other applications that enable data display and are used to perform the detailed analysis of the results such as STAR, another MacDonald Dettwiler product developed for DRDC Atlantic.

The SPPACS software suite consists of two types of software. One type is runtime executables that can be used to process DRDC Atlantic data files in a number of ways including data management and signal processing. Each program performs a specific function and the programs are designed so that they can be used in conjunction to perform more complex processing tasks. The software has proven to be very useful in simplifying data management and sonar processing tasks by providing a set of tools from which to build the necessary processing streams. These streams can be run from the command line or assembled into scripts to perform batch-processing tasks allowing for large amounts of data to be automatically processed. The second form of the software is a group of library functions that can be used by other programs to efficiently perform standard tasks. These library functions are extensively used by the runtime software, but can also used for other applications.

SPPACS is also supported by a set of signal processing libraries known at Fastest Fourier Transform in the West (FFTW) [3]. These free open-source libraries provide optimized signal processing functions helping to ensure that the SPPACS software runs as efficiently as possible, while providing a significant reduction in coding effort.

Before the spring of 2002 many of the custom software packages at DRDC Atlantic were under little to no configuration management. At this time, a co-op student was employed to migrate some of this software to the CVS repository where the management, development and installation of the software placed under the control of the DRDC Atlantic Software Control Project (SCP) would be standardized. SPPACS was part of this effort. The SCP version of SPPACS had a number of implementation issues and the functions used to support its configuration were in question at the time of contract award. A similar, but not identical approach to software management was chosen for SPPACS that maintains the spirit of the SCP.

Despite the work performed to continue to develop SPPACS and to place SPPACS under version control, formal test cases did not exist that allow users to confirm the correct operation of the software once compiled and installed. This lack of formal testing has occurred due to limited resource allocation and subsequent scope management on previous development contracts.

2. Work Overview

This section presents an overview of the tasked work and a summary of the work that was completed. The contract requirements from the Statement of Work (SOW) are presented along with an indication of the degree of success. Other significant work is also listed.

The requirements for this call-up are listed in Table 1. All requirements listed in the table were implemented with the following exception:

- ECHO_ENERGY_SPECTRA – It was found that this method of measuring the receive levels was not appropriate so Energy Time Integration (ETI) data was produced from a single spectra rather than a series of spectra. The analysis was then performed using existing STAR functionality. Some work was performed towards this task, but most of this work was related to infrastructure that can be used to implement other spectral analysis tools, when required.

A number of other enhancements, functions and utilities were developed that are not directly reflected in the requirements. These changes were made to improve the utility and reusability of the software or to address known issues, under authorization of the SA. As this work is incremental, parts of the work described below may have been started under other contracts. The additional work includes:

- The analysis displays were modified to allow the user to resize them at runtime. The widgets displayed in the window are updated when the user releases the mouse button.

- Window close events are now captured before the window exits. This allows the applications to cleanup dynamically allocated memory before they exit.

- A utility was written to generate HTML documents from IDL source files. The goal of this utility is to allow the API documentation to be extracted from the source files. This reduces the likelihood that the IDL code and the documentation describing it will get out of sync. The first version of this software offers a basic parsed view of the source code. Later versions will allow more sophisticated parsing (i.e. contents located on the left of the framed page as shown in Figure 6 will be collapsible, provide searches and source code links, etc.).

Table 1. *Contract Requirements*

REQUIREMENT TITLE	ASSOCIATED REQUIREMENTS
sp_spectra processing:	• The contractor shall generate SPPACS processing script(s) that process the provided raw data using sp_spectra to produce DRDC formatted .pwr files given the required processing information.
ECHO_ENERGY_SPECTRA:	• The contractor shall develop spectral analysis tool that measures and records the o Signal energy o Estimate of the noise energy under the signal o Signal duration o Signal frequency limits • The log entry shall also contain, at a minimum, the range and bearing between the provided source and receiver positions. • The measurement process shall allow the user to select a rectangular box defining the signal limits in time and frequency during analysis and to specify the extended range in time and frequency to use for the noise estimate prior to run time.
Suspend Analysis:	• The contractor shall implement a suspend analysis function that copies the current data to temporary variables, prints a short description of the data suspends processing to allow the user to manipulate and analyze the data manually. • This function shall be integrated into the Power Spectra and ETI analysis modules.
Automated Re-analysis:	• Is it possible that a software or process defect will invalidate the analysis results? The contractor shall implement an automatic reanalysis function that will use current log and data header information to feed the input parameters to the available analysis functions for reanalysis. This function will assume that the log entry format remains unchanged or that the parsers have been modified to accept both the new and old format.
Tactical Plot Integration:	• The tactical plot module and associated position and contact model shall be integrated into the Power Spectra analysis module. • Design changes required to update the Power Spectra module to the level of the ETI analysis module shall be incorporated.
Hardcopy Generation:	• The contractor shall implement a function that causes a display update with the output device set to an EPS (preferable) or PS file, as available. The output will be stored in a user-designated file. • This function shall be integrated into the Power Spectra and ETI analysis modules.

3. Digitize Data

A number of trial tapes exist in DAT format. A digital version of this data was required to facilitate analysis. This data was replayed on a Metrum recorder to produce analogue data and then digitized using the EADAQ system, producing DREA formatted time series files. These files were then transferred by ftp to //hokulea/bigdisk/canuk_data.

Separate files were created as follows:

- Each data filename shows the date and time that the original recordings were made.
- All of the 3G or larger files have data at the end of the file that is noise (some have 2 to 3 hours as they were done overnight - 12 and 13 hour recordings actually digitized for 15 hours).
- All other files were started and stopped when the file started/ended.
- Filename 30MAY99_111406.DAT is two recordings in one file.
- The data was sampled at 4K samples per second using the EADAQ.
- The following information was copied from the front panel of the Metrum recorder. These settings were used for all recordings:

Ch	Annotation
1	BOW HP 0dB, ARX BD1 CH1
2	BOW HP 12dB, ARX BD1 CH2
3	FIN HP 0dB, ARX BD1 CH3
4	FIN HP 12dB, ARX BD1 CH4
5	STERN HP 0dB, ARX BD2 CH1
6	STERN HP 12dB, ARX BD2 CH2
7	BEACON VOLTAGE 0.1V/V
8	IRIG B TIME CODE

SRC	IN(V)
1	2
2	2
3	2
4	2
5	2
6	2
7	10
8	5

PEAK	OUT(V)
1	2
2	2
3	2
4	2
5	2
6	2
7	2
8	2

4. Data Processing and Analysis

The data processing and analysis involved several tasks including:

- Conversion of Non-Acoustic Data (NAD) to STAR Format
- Processing trial and test data using SPPACS scripting
- Processing test and calibration
- Signal Level Measurement

4.1 NAD Formatting

NAD was required to annotate the received signal level measurements with range and bearing data. The source ship was RV2 and the target was RV4. NAD was required for both of these platforms. American Standard Code for Information Interchange (ASCII) based files with each platform's position over time were provided by the SA and they were converted to the STAR NAD format using the process described in, "Support to Multistatic FM Data Processing – The SPPACS / STAR Data Analysis Process and Toolset". Files were created for:

- Main trial file – provides trial area and time period information
- Source (RV2)
- Target (RV4)
- Receiver – set to match the RV2 position for all receivers (This provided the required geometric data from 'Echo' measurements.)
- Ping – links waveform processing to a source

4.2 Data Processing

The raw data files produced by EADAQ were processed using three methods to facilitate data assessment and analysis. This report will focus on the processing stream used in the data analysis. The reader can refer to the trial processing scripts for the details on the other processing streams. All data was byte-swapped to little-endian format prior to processing. All test cases were processed using the same processing stream as the trial data.

The three processing streams used were:

- Lofargram processing – this stream was used to resolve the individual ping types in a lofargram to aid visual recognition and analysis of the pulses of interest. It proved very useful in identifying various ping types, multi-path, saturation and a dead channel.

- Correlation processing – this stream was set for the waveform of interest and was used to identify the multiple signal arrivals. As expected, it provided much better resolution than the lofargram processing.

- Signal Level processing – this stream was used for the actual data analysis and is described in detail. The objective was to capture all significant received energy in one Fast Fourier Transform (FFT).

4.2.1 Signal Level Processing

The following processing was used to process the signal level data:

- 24K Fourier Transforms (FT) were created using sp_spectra with no window and 75% overlap. Bins 6142 to 8016 were passed to the next processing stage. This corresponds to a band from 1000 Hz to 1305 Hz and a 6.14s window. With an overlap of 4.61s one could be certain to capture the entire main pulse (4.5s long) in one processing window.

- After FFT processing the spectral magnitude was taken using sp_crossspectra.

- An ETI was then formed using sp_eti to integrate all energy in the band.

The 24K FT size was selected by processing the data using various FT sizes between 16K and 32K until the maximum output level stabilized showing that all required energy had been captured in the processing window.

The processing was tested by showing that the manually calculated receive levels were identical to the output of the signal processing.

4.3 Calibration and Testing Methodology

It was a requirement to validate both the signal processing and processing methodology. It is known that signal levels can be readily measured using the time domain, but there was some question about the corrections required to perform the same measurement in the frequency domain. The time and frequency domain signal levels were theoretically and numerically computed to validate the frequency domain processing methodology. The numerical computations were performed on synthetically generated data containing a perfect High Frequency Module (HFM) waveform.

4.3.1 Time domain energy

The time domain energy level is given by

$$ETI(t) \equiv \int_{t}^{t+T} V^2(\tau) d\tau \tag{1}$$

where t is the time, T is the integration time window and $V(t)$ is the signal voltage.

The signal voltage is given by:

$$V(t) \equiv v_{LSB} s(t) \tag{2}$$

where v_{LSB} is the voltage level of the least significant bit and $s(t)$ is the sampled signal.

Introducing F_s, the sampling frequency and N, the number of time samples in the integration time window, the discrete time domain expression of ETI is then given by:

$$ETI(t) = \frac{v_{LSB}^2}{F_s} \sum_{n=0}^{N-1} s_n^2 \tag{3}$$

Where $\{s_n\}$ is the time series of the normalized signal $s(t)$.

4.3.2 Discrete Fourier Transform Conventions

The FT can be described using two conventions. In this case one convention is used by IDL and another is used by SPPACS. Each conventions use in the calibration follows along with its application.

4.3.2.1 First Convention

The conventional definition of the Discrete Fourier Transform (DFT) $\{\hat{S}_k\}$ of the time series $\{s_n\}$, for an N sample window is given by

$$\hat{S}_k \equiv \sum_{n=0}^{N-1} s_n e^{-2i\pi \frac{nk}{N}} \tag{4}$$

$$s_n \equiv \frac{1}{N} \sum_{k=0}^{N-1} \hat{S}_k e^{2i\pi \frac{nk}{N}} \tag{5}$$

The fftw library called by SPPACS uses this definition.

When following this convention, Parseval's theorem is expressed in a discrete form using

$$\frac{\left(\sum_{k=0}^{N-1}\hat{S}_k^2\right)}{\left(\sum_{n=0}^{N-1}s_n^2\right)} = N \tag{6}$$

4.3.2.2 Second Convention

A second convention, used by IDL, is

$$\hat{S}_k \equiv \frac{1}{N}\sum_{n=0}^{N-1} s_n e^{-2j\pi\frac{nk}{N}} \tag{7}$$

$$s_n \equiv \sum_{k=0}^{N-1}\hat{S}_k e^{2j\pi\frac{nk}{N}} \tag{8}$$

Using the second convention, the discrete form of Parseval's theorem is given by

$$\frac{\left(\sum_{k=0}^{N-1}\hat{S}_k^2\right)}{\left(\sum_{n=0}^{N-1}s_n^2\right)} = \frac{1}{N} \tag{9}$$

4.3.3 Frequency Domain Energy

The calculations necessary to determine the energy contained in a signal in frequency domain is dependant on the convention used. The following subsections provide the equations necessary to define the ETI function for both conventions.

4.3.3.1 SPPACS Expression

Using the first DFT convention and Parseval's theorem, the expression for ETI in the frequency domain is given by

$$ETI(t) = \frac{v_{LSB}^2}{NF_s}\sum_{n=0}^{N-1}\left|\hat{S}_k\right|^2 \tag{10}$$

The sum over N can be reduced to a sum over $N/2$-1, using the DFT hermitian symmetry, which leads to the discrete frequency domain expression of ETI used in SPPACS

$$ETI(t) = 2\frac{v_{LSB}^2}{NF_s}\sum_{n=0}^{N/2-1}\left|\hat{S}_k\right|^2 \tag{11}$$

4.3.3.2 IDL Expression

The second DFT convention and Parseval's theorem leads to the frequency domain expression for the ETI function that was used in IDL calibration script and is given by

$$ETI(t) = \frac{Nv_{LSB}^2}{F_s}\sum_{n=0}^{N-1}\left|\hat{S}_k\right|^2 \tag{12}$$

4.3.4 Calibration

Calibration was performed by comparing the results of the three ETI calculation methods performed on the synthetic time series data. The ETI function was first calculated in both time and frequency domain, using equations (3) and (12) respectively. These tests are contained in the IDL script /bigdisk/canuk_data/perform_calibration.pro on hokulea). The third ETI method was calculated with SPPACS using equation (11). This equation is implemented in the processing stream generated for this contract.

The output from the three ETI computation methods were compared and identical numerical results were obtained for all cases showing that both the processing and methodology were valid.

4.4 Data Analysis

A number of customizations were used to support data analysis. A custom IDL script was generated from the MDA_template/eti_anal_template.pro file and placed in the trial's idlprog directory. The custom script was named eti_analysis.pro. The eti_analysis_constants.pro file was also copied to the trial's idlprog directory to allow manual override of the echo energy window definition for this trial. IDL scripts were also generated for lofargram viewing (gram_view.pro) and correlation ETI viewing (corr_analysis.pro). Both of these scripts were placed in the trial's idlprog directory.

Initial data analysis was performed using the file 30MAY99_111406_LE.eti. The waveform DWF004 was analyzed for all cases and the 0 dB gain channels for all three hydrophones (bow, fin and stern) were measured.

After analyzing the first three pulses it was found that a second waveform arrived at the receiver shortly after the DWF004 waveform, producing a double bump in the ETI. The lofargram viewer was used to confirm this that the first of the two bumps corresponded to DWF004.

Nine pulses were analyzed and recorded to a log file using the STAR's ETI echo measurement tool. The results were imported into Open Office and graphed. DRDC staff analyzed the rest of the data, as contract funds were insufficient to allow full analysis of the dataset.

5. Developed Software Components

The following sections describe the developed software components in detail. Each subsection is derived from the software documentation stored in the docs/reports subdirectory of the SPPACS distribution or in the documents subdirectory of the STAR distribution. These reports are based on a standard format that is modified as required for each program and serves as the technical documentation of the program. The standard outline for each report contains:

- A short description of the program and recent work performed
- A list of the program's functional requirements
- A description of the software design
- The software test description detailing the testing used
- The software test report detailing the results of testing

No new SPPACS utilities were developed under this contract. Existing utilities were utilized to complete the data processing requirement, as described in Section 4. Processing scripts were created to use the existing utilities to process the raw data.

Several enhancements were made to the STAR toolkit to support this call-up. They include suspend analysis, hardcopy generation and automatic re-analysis. The suspend analysis feature allows the operator to save a 'snapshot' of the data from the ETI or power spectra displays. The hardcopy generation feature allows the operator to write the ETI or power spectra display to a postscript (ps or eps) file. The power spectra display was upgraded to use the same event handling routines (higher level of abstraction than original routines), user start-up script (bootstrap) and the tactical plot display. A new utility was created to allow the operator to re-analyze data processing results. This may be required if there was a software or process defect present while the data was being processed.

5.1 Suspend Analysis

This feature allows the user to access the currently displayed data thereby providing maximum flexibility when processing trial data. It provides the user access to the header definitions and associated data displayed in the analysis displays (ETI, power_spectra, etc.). The user can manually manipulate the retrieved data by using built-in or custom IDL routines.

5.1.1 Requirements information

- The contractor shall implement a suspend analysis function that copies the current data to temporary variables, prints a short description of the data and suspends processing to allow the user to manipulate and analyze the data manually.

- This function shall be integrated into the Power Spectra and ETI analysis modules.

5.1.2 Software Design information

The suspend analysis feature consists of a reusable caching module, common block and access routines. It stores one or more data handles, more of which can be added to the cache at runtime. When the user activates the suspend feature the caching module retrieves a copy of the data structures referenced by the data handles stored in its settings structure (as illustrated in Figure 1). A new structure is created at runtime using the retrieved structures and it is stored in its own data handle.

The user can save the display data by clicking the Suspend Analysis button located on the power spectra or ETI display. The cached data persist in memory, referenced by the cache data handle, until the application exits or the user activates the suspend analysis feature again at which point the data is over-written with new data.

Figure 1. *Suspend Analysis – Save Analysis Data*

The stored data can be accessed with the get_suspend_data function. This function uses the analysis cache handle stored in the common block to retrieve the cached data (as shown in Figure 2). The data is then returned to the user at which point the user can view and/or continue processing the data.

Once retrieved, users can plot data, change the number of plots displayed, etc. as they would normally. As long as the analysis window is not closed, the user can enter a valid command or select an update 'u' to redisplay the original data and continue processing.

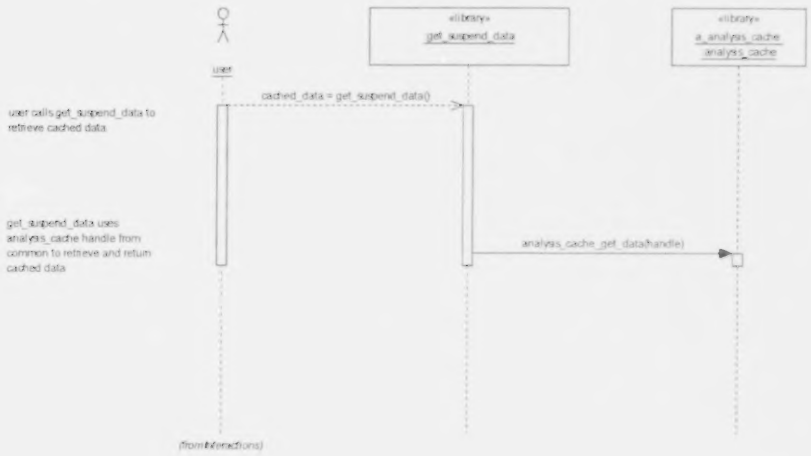

Figure 2. *Suspend Analysis - Access Analysis Data*

5.1.3 Test Description

The following test cases are used to test the suspend analysis feature:

- Suspend Analysis on ETI display: Load the ETI display, activate the suspend analysis feature, retrieve the stored data using get_suspend_data(), retrieve the original data by de-referencing the appropriate data handle. The data should be the same.

- Suspend Analysis on power spectra display: Load the power spectra display, activate the suspend analysis feature, retrieve the stored data using get_suspend_data(), retrieve the original data by de-referencing the appropriate data handle. The data should be the same.

- Regression tests: Automated regression tests can be run by executing analysis_tools/MDA_tactical_database/analysis_cache_test.pro. These test cases test the lower level routines.

5.1.4 Test Report

Table 2. Test Results – Suspend Analysis

TEST CASE	RESULTS (PASS/FAIL)	NOTES
Suspend Analysis on ETI display	Pass	
Suspend Analysis on power spectra display	Pass	
Regression tests	Pass	

5.1.5 User Instructions

A Suspend Analysis button was added to the ETI and power spectra displays. A dialog is present to the user to describe the data stored in the cache. After the data has been saved the user can access it for further processing using the get_suspend_data function.

IDL> data = get_suspend_data()

The returned data is normally a data structure that can be assessed using

IDL>help, data, /structure

Substructures can be analyzed in a similar manner. ETI and power spectra data consists of a header (hdr) structure and a data array.

5.2 Automated Reanalysis

The purpose of the automate reanalysis routines is to allow the user to reanalyze log files (Power Spectra or ETI). This can be done to correct log corruption or inaccuracy due to processing or software defects. Reanalysis can also be used to upgrade analysis logs to newer formats or to automatically generate comparative analysis on the same dataset. For example, a user may wish to compare the performance of two processing algorithms.

The reanalysis routines are run from the command line and require no user interaction once started. The reanalysis program uses the template program approach for settings and initialization[1]. As the routine runs it initializes the database, with the user provided trial file. After initialization of the database the software contains all of the information from the log files to be analyzed. The log files must be located in the <trial>/NAD/log directory in order to

[1] For more information see 'IDL user scripts.doc'.

be reanalyzed. Each log entry is analyzed and written to the new log file that was specified by the user.

5.2.1 Requirements

- It is possible that a software or process defect will invalidate the analysis results. The contractor shall implement an automatic reanalysis function that will use current log and data header information to feed the input parameters to the available analysis functions for reanalysis.
- This function will assume that the log entry format remains unchanged or that the parsers have been modified to accept both the new and old format.
- Trial File – Specifies parameters needed to initialize the tactical database.
- Log File Name – Specifies the path and name of the new log file to be generated from the reanalysis. Repeated use of the same log file will result in message being append to the end of the file.

5.2.2 Software Design

Calls from the main function initialize the modules needed for reanalysis (as shown in Figure 3). Once all of the modules have been initialized the reanalysis process begins. Specifically the main function initializes the tactical database using the trial file to find all NAD. Next all of the log entries to be reanalyzed are read from the database into local variables. A dummy initial request is generated to allow initialization of the supporting modules. The supporting modules are then initialized and the reanalysis modules begin reanalyzing the log entries. Each entry is written to the user specified log file.

5.2.2.1 Modules

The Reanalyze_eti_log and the Reanalyze_power_spectra_log control the flow of the programs. Each utilizes the analyzer modules listed below. Each analyzer module is responsible for updating the models and then invoking the existing analysis algorithms. The same software functions that were used to generate the original logs are used in the reanalysis.

- Reanalyze_eti_log – Controls the flow of the program.
- Reanalyze_power_spectra_log - Controls the flow of the program.
- Main_blast_analyzer – Reanalyzes all of the main blast entries read from the database.
- Echo_analyzer - Reanalyzes all of the echo entries read from the database.

- Echo_energy_analyzer – Reanalyzes all of the echo energy entries read from the database.
- No_contact_analyzer – Reanalyzes all of the no contact entries read from the database.
- False_alarm_analyzer – Reanalyzes all of the false alarm entries read from the database.
- Echo_energy_spectra_analyzer - Reanalyzes all of the echo energy spectra entries read from the database.
- Analyzer_utility – Provides common functionality for both reanalyze_eti_log and reanalyze_power_spectra_log.

trial file and log file checks
instantiate tactical_database
get all logs from tactical database
initialize supporting modules
call analysis routines

Figure 3. *Reanalysis Activity Diagram*

5.2.3 Test Description

The following test cases were used to test the automated reanalysis program:

- Log file comparison test case: Run the program with the path and name of a valid trial file and new log file. Compare the new log file with the old log file with the diff command. Assuming the initial log file was valid there will be no difference between the new and old log file.
- Bad path/filename test case: Provide a bad filename or path to the trial file and new log file to be created. An error is printed and the program exits.

5.2.4 Test Report

Table 3. Test Results – ETI and Power Spectra

TEST CASE	RESULT (PASS/FAIL)	FAILURE DETAILS/ NOTES
Log file comparison test case	Pass	
Bad path/filename test case	Pass	

5.2.5 User Instructions

The main function can be invoked from a script called eti_reanalysis_template.pro or pwr_spec_reanalysis_template.pro found in the MDA_template directory. Alternatively both programs can be run from the command line by providing the function name along with the necessary parameters. The first parameter must be the path and name of the trial file. The second parameter must be the path and name of a log file.

5.3 Hardcopy Generation

The purpose of hardcopy generation is to allow the user to write the contents of the ETI and power spectra window to a postscript file in either ps or eps format. The ability to print a screen capture, to file, also exits, but this only produces a bitmap and not an editable file.

5.3.1 Requirements

- The contractor shall implement a function that causes a display update with the output device set to an EPS (preferable) or PS file, as available. The output will be stored in a user-designated file.

- This function shall be integrated into the Power Spectra and ETI analysis modules.

5.3.2 Software Design

The hardcopy generation feature is invoked from the ETI or Power Spectra analysis windows by pressing the Save to Postscript button. The postscript button will generate an event that is captured by the window. This event starts the initialization of the display output module that is responsible for switching the output device and writing the screen to postscript (as shown in Figure 4).

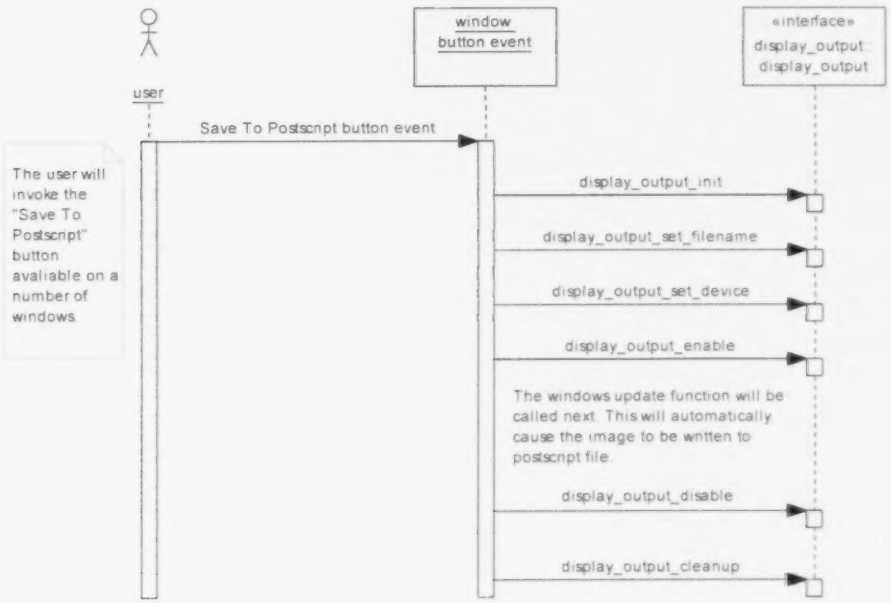

Figure 4. *Hardcopy Generation Sequence Diagram*

5.3.3 Test Description

The following test cases are used to test writing windows to postscript. The output files were reviewed with gimp to ensure correctness.

- Write to postscript (.ps, .eps) test case: Press the "Save To Postscript" button and select a valid .ps/.eps file and press OK. Open the file with gimp and compare it with the current window. The postscript file will be identical to the current window except the back ground is white and the test is black.

- Capture screen test case: Press the "Capture Screen" button and select a valid .ps/.eps file and press OK. Open the file with gimp and compare it with the current window (they will be the same)

- Invalid file extension test case: Follow the steps in the write to postscript test case but select a file extension other than .ps/.eps. A warning message is displayed stating that you did not pick a valid file.

- Invalid file name test case: Follow the steps in the write to postscript test case but do not select a filename. A warning message is displayed stating that you did not pick a file.

- Cancel postscript test case: Follow the steps in the write to postscript test case but click the cancel button instead of the OK button. The operation should cancel return control to the displays event loop and wait for user interaction.

5.3.4 Test Report

Table 4. *Test Results – Hardcopy Generation (ETI, Power Spectra, Tactical Plot, BBTR)*

TEST CASE	RESULT (PASS/FAIL)	FAILURE DETAILS/ NOTES
Write to postscript (.ps, .eps) test case.	PASS	
Capture screen test case	PASS	
Invalid file extension test case	PASS	
Invalid filename test case	PASS	
Cancel postscript test case	PASS	

6. Software Maintenance and Design Improvement

The work described in this section was necessary to ensure the long-term viability and reusability of the toolkit. The SA authorized the work described in this section, as the contract work was progressing ahead of schedule and these issues were troublesome to effective use of the software.

6.1 Window Resizing

The analysis displays were modified to allow the user to resize them at runtime. The user can click on the edge of the window and drag it to increase or decrease the size of the window. The widgets displayed in the window will be updated when the user releases the mouse button.

6.1.1 Software Design information

Most of the infrastructure had been developed under previous contracts. This enhancement focused on creating a reusable solution that could be integrated easily with all of the displays.

Some set up was and will be required for each display before a window resize event can be handled. Create the widget_base using /TBL_SIZE_EVENT. This will send any window resizing to the event loop. After creating the widget_base and the addition of buttons, menus (etc.) draw the canvas. Calculate the size of the canvas minus the buttons and menu. Call widget_control, base_id, tlb_get_size=canvas_size to get the initial size of the canvas.

After the display has been initialized it will wait for user interaction. When a resize event is trapped in the event function perform another widget_control tlb_get_size. Get the size of the resized canvas by using the widget_control procedure with the tlb_get_size keyword. Calculate the difference between the initial canvas size and the new canvas size. Add the initial canvas size and the difference together to get the size of the new window. Store the new size for later use. Make a call to widget_control, canvas_id, xsize=new_xsize, ysize=new_ysize.

6.1.2 Test Description

The ETI, power spectra and tactical plot displays were manually tested to ensure that they would resize correctly.

6.2 Capture Window Close

It is often desirable or necessary to capture the close event generated from the close button built into the IDL frame of a widget. This is useful because it allows the software to execute cleanup routines before the program exits.

6.2.1 Software Design information

The solution to this problem is to create your widget and then call widget control with the /TLB_KILL_REQUEST_EVENTS keyword set. Setting this keyword will allow the close event to be captured by your event loop function created by XMANAGER. Check for a close event with:

if(TAG_NAMES(event, /STRUCTURE_NAME) EQ 'WIDGET_KILL_REQUEST') then begin ...

6.2.2 Test Description

The tactical plot display was manually tested to ensure that the window close button was captured properly. After the window is closed the help,/heap command was used to check for dynamic memory that has not been de-allocated.

6.3 IDL API Document Generation

A utility was written to generate HTML documents from IDL source files. This utility will recursively search a directory for all IDL source files and generate HTML documentation for each. It will create a framed index page.

6.3.1 Requirements Information

- Will parse IDL source files.
- Will automatically extract structure, function and procedure definitions.
- Will generate links on the html file to other IDL source files that the parsed file depends on.
- Will parse tags that allow the developer to add comments, extra code, constants, etc.
- Will recursively search the root directory and all sub directories for IDL source files.
- Will allow the user to exclude files or directories.

6.3.2 Software Design information

The IDL source parser consists of two PERL scripts and a shell script. parser.pl will create a single HTML file from a single IDL source file. idl_src_parse.pl will parse the current working directory and all sub-directories.

parser.pl will automatically parse structure, function, procedure definitions, generate a list (links) to dependencies (from include files and constants includes). It can run from the command line:

usage: parser.pl [--html_base=BASE_DIRECTORY][-f=exclude_file1,exclude_file2 ...][-l=LIBRARY_DESCRIPTION_PAGE]

- --html_base specifies the location to write the html files to (default is ./html).
- -f specifies files or directories to exclude from parsing (by default the script will not parse CVS directories)
- -l specifies the cover page to display. (uses a blank page by default).

Anything can be included (code, comments, etc.) by including a starting ';+' and ending ';-' (as shown in Figure 5). A horizontal rule or separator between sections of text can be added by including a ';!' in the source code.

```
; $Id: data_handle.pro,v 1.8 2004/01/30 15:15:03 glessing Exp $

;!
;+
; This library is used by ADT to encapsulate data with a integer
; handle (long).
;
; Note: data_handle_init must be called once when the application
; starts and data_handle_cleanup must be called once when the
; application exits.
;-

pro data_handle_entry__define
  struct = {data_handle_entry, handle:0L, settings:ptr_new()}
end


pro init_data_handle, NUMBER_HANDLES=number_handles
;+
; allocates space in common block for handles
;
; number_handles (int)  max number of data handles available (see
;                       data_handle_constants for default
;                       number_handles (default_handle_size)
;
;-
  @data_handle_constants
  common data_handles, handle_count, handles_size, handles

                          ; set initial handle number
                          ; new handles are always handle_count
  handle_count = default_start_handle

                          ; allocate handles array
  if(keyword_set(number_handles)) then begin
    handle_size = number_handles
  endif else begin
    handles_size = default_handle_size
  endelse

  handles = replicate({data_handle_entry},default_handle_size)
  handles[*].handle = invalid_handle
end
```

Figure 5. *Sample IDL with Tags*

Idl_src_parse.pl will parse all IDL source files in a directory and all of its sub directories. The script will create a framed page with a table of contents on the left and the currently viewed page on the right (will start on a default or user supplied cover page) (as shown in Figure 6).

Figure 6. idl_src_parse HTML Output

6.3.3 Test Report

The parser was executed on several files from the STAR distribution. The HTML outputs were manually checked against the original source files for errors.

6.3.4 User Instructions

A shell script has been created to make the parser easier to execute:

1. Set environment variables: IDL_PARSE_BASE to $BASE_DIR/analysis_tools/MDA_IDL_doc and add $BASE_DIR/analysis_tools/MDA_IDL_doc to your PATH.

2. Execute idl_src_parse in the directory that you wish to parse.

7. Data Analysis Recommendations

A number of significant enhancements were identified during the data analysis work. These enhancements would make the analysis displays easier to use and as a result enable the user to focus more attention to data analysis and/or decrease data analysis time. The following recommendations are made for STAR ETI tool enhancement:

- Allow synchronized viewing of both the lofargram and ETI data to support pulse classification
- Provide a 'jump to time' option to quickly navigate past large gaps in significant data
- Convert the data base query and data stream processing to use pointers rather than data copies to speed processing
- Allow the user to edit the parameters for "ping length" and "tweak" from the main display, so that the user don't have to edit and recompile the routine each time
- Change the font size to a bigger font for high-resolution screens or allow the user to change the font
- Use a pop-up error box when an important error such as can't write to log file occurs

8. Maintain Configuration Management

STAR and SPPACS are maintained using CVS. The most recent release version is maintained and bug fixes are applied to that version, as required, ensuring that a stable release is always available. Simultaneously, software enhancements are applied to the development version and bug fixes are merged with this version. Once a call-up nears completion or a release of the software is otherwise required a new release version is branched off of the development stream for final integration, release testing and delivery.

EADAQ and the Noise Monitoring Software are maintained in a similar manner using the RAZOR Configuration Management tool.

STAR release 3.0 (tag release_3_0) and SPPACS release 2.0 (tag release_2_0 – only bug fixes) were created under this contract to serve as a baseline while the analysis work was being conducted. This allows development to continue on the trunk of the distribution. The trunk usually contains newly implemented software, which may not be stable enough to allow for operational use.

9. Track Software Issues

Defect Tracking Systems allow users to keep track of outstanding bugs in their product effectively. STAR and SPPACS issue tracking is performed using a Web accessible tool called Bugzilla. This tool can be accessed using a secure web interface at https://star.iotek.ns.ca. Once the appropriate security procedures, detailed on the web page, have been followed, users and developers can use this site to add, view or modify issues related to the software packages. If required, the EADAQ and Noise Monitoring Software packages can also be added to the site.

A break down of the current issues for the STAR and SPPACS distributions is shown in Table 5. The total number of unresolved issue in shown in the NEW/ASSIGNED/ REOPENED column. The total number of opened issues is broken into to classes of severity. Issues classified as BLOCKER/CRITICAL/MAJOR are issue that should be addressed in the short term. Blockers are always addressed immediately to ensure that the user community can continue with their work. Issues classified as NORMAL/MINOR/TRIVIAL are issues that can be dealt with in the long term.

Table 5. *Distribution Issue Summary (18/03/2004)*

PRODUCT	NEW/ASSIGNED/ REOPENED	BLOCKER/CRITICAL/ MAJOR	NORMAL/MINOR/ TRIVIAL
SPPACS	35	1	34
STAR	34	2	37

The following gives a more detailed description of the SPPACS BLOCKER/CRITICAL/ MAJOR column:

- Issue # 211 (critical) **pre-built fftw library will only work on redhat**. The work around for this issue is to download fftw, build it on the target platform and modify the FFTW settings in <SPPACS_ROOT>/sppacs/make.common.

The following list gives a more detailed description of the STAR BLOCKER/CRITICAL/ MAJOR column:

- Issue # 193 (critical) **tacplot does not cleanup before exiting**. The work around for this issue is to run heap_gc after the analysis window has closed.
- Issue # 107 (critical) **problems capturing close button**. A solution exists for this but has only been incorporated into the tactical plot.

Bibliography

1. Joe Hood and Brad Glessing, *The SPPACS / STAR Data Analysis Process and Toolset -Analysis process and technical manual, draft.*

2. Joe D. Hood, and Brad Glessing, *SPPACS Maintenance and Enhancement*, DRDC Atlantic CR 2004-074, April 2004.

3. Matteo Frigo and Stephen G. Johnson, *FFTW: An Adaptive Software Architecture for the FFT*, ICASSP conference proceedings (vol. 3, pp. 1381-1384) 1998

List of symbols/abbreviations/acronyms/initialisms

API	Application Program(ming) Interface
ASCII	American Standard Code for Information Interchange
BBTR	Broadband Bearing Tracker
CVS	Concurrent Versioning System
DFT	Discrete Fourier Transform
DND	Department of National Defence
DRDC	Defence Research and Development Canada
DREA	Defence Research Establishment Atlantic
DRP	Document Review Panel
EA	Enterprise Architect
EADAQ	Environmental Acoustic Data Acquisition System
EPS	Encapsulated Postscript
ETI	Energy Time Indicator
FFT	Fast Fourier Transform
FFTW	Fastest Fourier Transform in the West (C subroutine library)
FM	Frequency Modulated
FT	Fourier Transform
HFM	High Frequency Module
HTML	Hyper Text Markup Language
IDL	Interactive Data Language

IMPACT	Integrated Multistatic Passive Active Concept Testbed
LFA	Low Frequency Active
NAD	Non-Acoustic Data
PC	Personal Computer
PS	Postscript
RISO	Regional Individual Standing Offer
SA	Scientific Authority
SCP	Software Control Project
SOW	Statement of Work
SPPACS	Signal Processing Package
STAR	Software Tools for Analysis and Research

Distribution list

Document No.: DRDC Atlantic

Internal List

6 DRDC Atlantic LIBRARY FILE COPIES
1 Mario Boutin
1 James Theriault
1 Brian Maranda
1 Nicole Collison

10 TOTAL LIST PART 1

LIST PART 2: DISTRIBUTED BY DRDKIM 2-2-5

1 NDHQ/DRDKIM2-2-5

1 TOTAL LIST PART 2

11 TOTAL COPIES REQUIRED

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

MacDonald Dettwiler and Associates Ltd.
Suite 60, 1000 Windmill Road, Dartmouth, NS, B3B 1L7

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable).

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Final Report for Data Processing and Analysis Support to Measure Active Signal Levels

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Joe Hood, Brad Glessing, Olivier Beslin, Derek Jury

5. DATE OF PUBLICATION (month and year of publication of document)	6a. NO. OF PAGES (total containing information Include Annexes, Appendices, etc).	6b. NO. OF REFS (total cited in document)
April 2004	43 (approx.)	0

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

CONTRACTOR REPORT

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

Defence R&D Canada – Atlantic
PO Box 1012
Dartmouth, NS, Canada B2Y 3Z7

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant).	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).
11cm01	W7707-03-2304

10a ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
	DRDC Atlantic CR 2004-075

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(x) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

This report documents the work done to convert raw multistatic trial data to received signal level measurements. In this case the signals were received at three hydrophones placed on the target submarine during Q246. The work was broken down into a number of tasks with several tasks involving the generation of reusable software components and analysis processes, allowing future work to be performed more efficiently. The received signals were digitized using the Environmental Acoustic Data Acquisition System (EADAQ) and processed using existing components from the Signal Processing Package (SPPACS) software suite. The Software Tools for Analysis and Research (STAR) Suite was then used to perform data analysis including test, calibration and measurement tasks. Analysis logs were imported into Microsoft Excel for archiving and final analysis of the results. Considerable efficiency was realized by using common databases, software and processes that Defence Research and Development Canada (DRDC) plans to reuse to perform similar tasks in the future.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

Underwater Acoustics
Data Analysis
Software